

OP1167-US

Prior Art Reference:

Japanese Patent Laid-Open Publication No. 10-222541

Laid-Open Date: August 21, 1998

Title: METHOD AND APPRATUS FOR PRELOADING WEB PAGE
CORRESPONDING TO LINK SPECIFIED IN HTML

Patent Application No. 10-4436

Filing Date: January 13, 1998

Priority Application No. 08/785912

Priority Date: January 21, 1997

Country: U.S.A.

Applicant: ID No. 390009531

INTERNATIONAL BUSINESS MASCHINES CORPORAITON
New York, U.S.A.

Inventor: Michael John Allen
New York, U.S.A.

Partial Translation:

[ABSTRACT]

[Object] To provide a method and system for immediate user access to another web page referred to by a link on a web page which a user is currently viewing.

[Solving means] A web browser loads and displays a web page, and reads a html of the web page and identifies an entry of a link marked as an object to be preloaded. Next, the web browser preloads automatically another html corresponding to the link, and a related graphics file to a storage device. The automatic preload is executed without the user selecting the link. If the user subsequently selects the link, the corresponding html and the related graphics file can be used from a local storage device, thereby there is little delay in taking the html and the related graphics file.

[DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS]

[0008]

Next, the drawings will be described in detail. The same reference numeral shows the same element throughout. Fig. 1 shows a client computer according to the

present invention, which is generally indicated by reference numeral 10. The client computer 10 comprises a web browser program 12 and a storage device 14 (RAMs and/or disks) for storing web pages. The web browser 12 provides a user interface, obtains a web page which a user requests by communicating with a server, and controls display of the web page to the user. The client 10 is connected to a WWW 16 through communication lines 18, such as modems and telephone lines or optical fiber lines. The WWW comprises various servers, such as a server 20. The server 20 has a web page access feature program 22 for gaining access to a web page of a disk 24 or requesting a web page from another server 26 for the client 10. In an example shown in the drawing, the client 10 communicates with the server 20 by using a http access system.

[0009]

When the server 20 supplies the client with a web page, the web page is defined by a html file built in http communication. As described in detail hereinafter, the html according to the present invention specifies a certain web page linked to the html as an object to be preloaded. On loading the html and displaying the corresponding web page, the web browser starts preloading the html corresponding to the specified link and graphics to the storage device 14. If the user subsequently selects one of these preloaded web pages through each of the links on the web page currently displayed, the web browser can take immediately and display the web page in the local storage device 14.

[0010]

The followings are examples of the html according to the present invention which are designed by a user.

```
33. <html>
35. <head>
37. <title>Patent Test Page</title>
39. </head>
41. <body>
43. <h3>Test Links</h3>
45. <ul>
47. <li><a href="http://www.test.com">Catalog</a>
49. <li><a href="http://www.test1.com">PRIORITY=1>Vaccum</a>
51. <li><a href="http://www.test2.com">PRIORITY=10>Carpet Shampooer</a>
53. </ul>
55. </body>
```

"<Html>" of the first entry 33 shows that this file is a html. "<Head>" of the next entry 35 shows a page header. The next entry 37 starts at a tag "<title>" showing that the subsequent text is a print job name and a title of a web page used for a quick list. The next entry 39 is a tag "</head>" showing the end of the page header. The next entry 41 shows a body of the html page. The next entry 43 shows a heading. The next entry 45 shows a list without numbers. The next entry 47 starts at meaning that it is an element of the list and then comprises a tag "<a>" meaning that a user can go to the specified URL, "www. test. com " by clicking the subsequent text "Catalog". Though this is an example and does not restrict the scope of the embodiments, "Catalog" web page is a catalog of a maker owning the currently viewed web page, which is written in text. The web page comprises only text and does not comprise graphics, therefore, it takes a short time to load the web page and the designer of the web page did not recognize the necessity of specifying the link as an object to be preloaded. Therefore, the entry 47 is a standard entry of html of the prior art. The next entry 49 also starts at meaning the element of the list and then comprises a tag "<a>" meaning that the use can go to the specified URL "www.test1.com" by clicking the subsequent text "Vacuum". Though this is an example and does not restrict the scope of the embodiments, the "Vacuum" web page comprises a photograph of the vacuum cleaner sold by the maker and other information thereon. Therefore, it may take quite a long time to load the web page and the designer of the web page recognized the necessity of specifying the web page as an object to be preloaded. According to the present invention, the entry 49 also comprises a field "PRIORITY = 1" meaning that a html of URL "www. Test1.com" must be preloaded as soon as the web page corresponding to the above-mentioned html is displayed. The preloading is executed automatically without the user requesting it or selecting the link corresponding to the entry 49. The next entry 51 also starts at meaning the element of the list and then comprises a tag "<a>" meaning that the user can go to the specified URL " www. test2. com " by clicking the subsequent text "Carpet Shampooer". Though this is an example and does not restrict the scope of the embodiments, the "carpet shampooer" web page comprises a photograph of the shampooer sold by the maker and other information related thereon and it may take quite a long time to load the web page. Therefore, according to the present invention, the entry 51 also comprises a field "PRIORITY=10" meaning that a html of URL " www. test2. com" must be preloaded as soon as the web page corresponding to the above-mentioned html is displayed. The preloading is executed automatically without the user requesting it or selecting the link corresponding to the

entry 51. As described in detail hereinafter, order of each html to be preloaded is determined by a priority level. The next entry 53 shows that it is the end of the list without numbers. The last entry 55 shows that it is the end of the html.

[0011]

In Fig. 2, a process by a browsing thread 60 in the web browser program 12 is shown. In step 62, the browsing thread receives the web page selected by the user through a URL or a link. For the purpose of description, it is assumed that the first selection in step 62 is done by the URL and that the URL indicates the above-mentioned html. In response to that, the browsing thread adds the URL to a dynamic load waiting line 63 (Fig.1) (step 64), and sends a message for instructing to process the dynamic load waiting line immediately to a loading thread 66 shown in Figs. 3 and 4 (step 65).

[0012]

Figs. 3 and 4 construct flow charts of the loading thread 66. When receiving a message from the browsing thread 60 (step 82), the loading thread first determines that the message does not relate to a process of the preload waiting line 73 (Fig.1), but a process of the dynamic load waiting line (determinations 84 and 130). The loading thread, next, determines whether the web page selected by the user currently exists in the storage device 14 as a result of the previous preload operation (104 determination). (The preload operation will be described hereinafter.) If the web page selected by the user exists, the loading thread takes the web page from the storage device 14 and displays it (step 106). The storage device 14 is a local storage device, therefore, the operation is executed quickly. However, if the web page selected by the user does not currently exist in the storage device 14, the loading thread determines whether a web page which is currently in process of being preloaded in accordance with the preload operation exists (determination 110). If the web page does not exist, the loading thread requests the web page selected by the user from the server, loads the web page selected by the user to the storage device 14, and displays the web page selected by the user (step 112). The preload operation returns to determination 110. If the web page which is currently in process of being preloaded in accordance with the preload operation exists, the loading thread determines whether the web page is the web page selected by the user (determination 113). If determined affirmative, the loading thread completes loading the web page selected by the user and displays the web page selected by the user (step 114). However, if a web page other than the web page selected by the user is currently in process of being preloaded, the loading thread stops preloading the other web page (step 120), and then requests the web page

selected by the user from the server, loads and displays the web page selected by the user (step 112). After any one of steps 106, 112 and 114, the loading thread notifies the browsing thread of success in loading and displaying the web page in the dynamic load waiting line (step 124).

[0013]

When receiving notification of display from the loading thread (step 67), the browsing thread reads a html of the currently displayed web page, finds all the links existing in the currently displayed web page and having a "priority" flag for instructing to preload (step 68). In the above-mentioned html, two such links are defined by the entries 49 and 51. However, the browsing thread first encounters a link defined by the entry 47, which is not marked as an object to be preloaded. Therefore, when encountering the entry 47, the browsing thread detours all the preload operations from the determination 70, moves to the determination 72, and from the determination 72, returns to the step 68 and checks the next entry 49. During the repetition of this operation, the browsing thread 60 encounters the entry 49 marked by the "PRIORITY" flag for instructing to preload (determination 70). As a result, the browsing thread adds a URL in the entry 49 at a position corresponding to a priority level of the preload waiting line 73 (step 74). The higher the priority level is, the closer to the head of the waiting line the URL is positioned. During the next repetition of the steps 68 and 70, the browsing thread encounters the entry 51 also comprising the "priority" flag for instructing to preload and the browsing function thereof adds a URL in the entry 51 to the preload waiting line 73 (step 74). In an example as shown in the drawings in which the entry 51 is marked "priority 10" and the entry 49 is marked "priority 1", the URL of the entry 51 is positioned ahead of the URL of the entry 49 in the preload waiting line. The browsing thread completes checking all the links in the above-mentioned html, therefore, the browsing thread notifies the loading thread 66 to process the preload waiting line immediately (step 76).

[0014]

The loading thread 66 receives a message for notifying to process the preload waiting line from the browsing thread (step 82 and determination 84), and in response to this, determines the first URL in the preload waiting line (determination 86). Next, the loading thread requests the corresponding html and the related graphics from the server. Even while the loading thread is receiving and loading the html and the related graphics (step 88), the loading thread checks the message from the browsing thread periodically. If the loading thread does not receive the message

(determination 90), the loading thread continues to load. That is to say, the loading thread determines whether further data of the current html and the related graphics to be loaded exists (determination 92). If the further data exists, a loop returns to step 88 and the loading thread loads the next part of the html or the graphics related thereto. However, if the loading thread has loaded the last portion of the current html and the graphics related thereto, the loading thread determines whether another URL exists in the preload waiting line (step 94 and determination 86). The loop returns to determination 86, and the loading thread reads the other URL. In the above-mentioned example in which the displayed web page comprises two web pages to be preloaded, at least two entries exist in the preload waiting line.

[0015]

The loop returns to determination 90. When the user selects a link and the loading thread receives a message for instructing to display the link immediately from the browsing thread (determination 90), the loading thread determines whether the user selects the web page which currently exists in the storage device 14 (determination 104). If determined affirmative, the loading thread takes the web page from the storage device 14 and displays it (determination 106). However, the web page selected by the user does not currently exist in the storage device 14, the loading thread determines whether a web page which is being loaded in accordance with the preload operation exists (determination 110). If the web page does not exist, the loading thread requests the web page selected by the user from the server, loads the web page selected by the user to the storage device 14, and displays the web page selected by the user (step 112), and then returns to determination 110. The web page which is currently in process of being preloaded in accordance with the preload operation exists currently, the loading thread determines whether the web page is the web page selected by the user (determination 113). If determined affirmative, the loading thread completes loading the web page selected by the user and displays the web page selected by the user (step 114). However, if a web page other than the web page selected by the user is currently in process of being preloaded, the loading thread stops preloading the other web page (step 120), requests the web page selected by the user from the server, and displays the web page selected by the user (step 112).

[0016]

After steps 106, 112, or 114, the loading thread notifies the browsing thread that the web page which is selected by the user and should be displayed immediately is displayed. The loading thread also determines whether any page which has not yet preloaded exists in the preload waiting line (step 120). If any, the loading thread

continues the process of determination 92. If any page does not exist, the loading thread is terminated.

[0017]

In accordance with the above-mentioned matters, the web browser according to the present invention is disclosed. However, various modifications and replacements can be worked without deviating from the scope of the present invention. If desired, for example, a function enabling the user to select optionally to completely stop preloads can be also provided for the web browser. Or, a function enabling the user to select optionally to stop preloads of which the priority levels are low, for example, to stop all the preloads of which the priority levels are under 5, can be also provided for the web browser. Therefore, the present invention is disclosed by the examples by which do not restrict the scope of the present invention. Claims defining the scope of the present invention are to be referred to.

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10-222541

(43) 公開日 平成10年(1998)8月21日

(51) Int. Cl.⁶ 識別記号
G 0 6 F 17/30
13/00 3 5 4

F I
G 0 6 F 15/419 3 2 0
13/00 3 5 4 A
15/40 3 1 0 F
3 8 0 D
15/403 3 8 0 A

審査請求 未請求 請求項の数 1 4

OL

(全 1 1 頁)

(21) 出願番号 特願平10-4436

(22) 出願日 平成10年(1998)1月13日

(31) 優先権主張番号 08/785912

(32) 優先日 1997年1月21日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72) 発明者 マイケル・ジョン・アレン

アメリカ合衆国13760 ニューヨーク州
エンディコット エクセター・ドライブ 5

(74) 代理人 弁理士 坂口 博 (外1名)

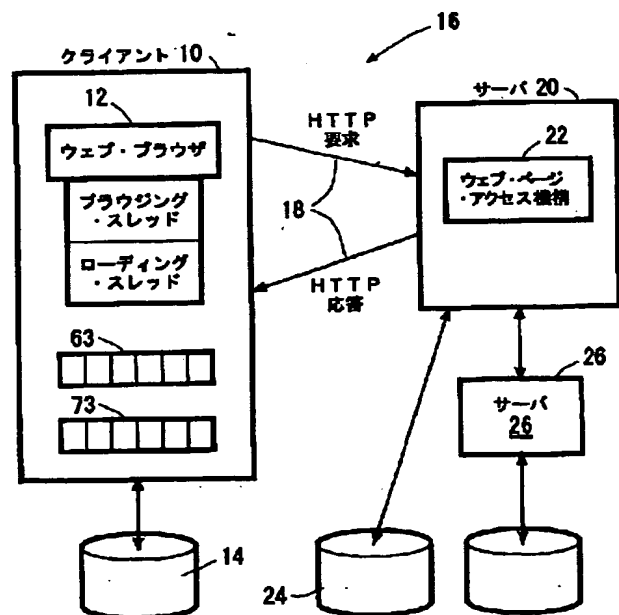
最終頁に続く

(54) 【発明の名称】 h t m l 中で指定されたリンクに対応するウェブ・ページの事前ロード装置および方法

(57) 【要約】

【課題】 現在ユーザが見ているウェブ・ページ上のリンクによって参照された別のウェブ・ページへのユーザのアクセスを迅速におこなう方法およびシステムを提供する。

【解決手段】 ウェブ・ブラウザは、ウェブ・ページをロードし、表示させ、このウェブ・ページの h t m l を読んで、事前ロードの対象として印を付けられたリンクのエントリを識別する。次いで、ウェブ・ブラウザは、このリンクに対応する別の h t m l および関連グラフィックス・ファイルを記憶装置に自動的に事前ロードする。この自動事前ロードは、ウェブ・ブラウザのユーザがリンクを選択しなくても実行される。ユーザが続いてそのリンクを選択した場合には、対応する h t m l および関連グラフィックス・ファイルが局部記憶装置から使用可能となり、そのため、h t m l および関連グラフィックス・ファイルの取り込みにおける遅延がわずかになる。



【特許請求の範囲】

【請求項1】htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する手段と、事前ロードの対象として印を付けられたリンクの識別に
10 応答して、前記リンクに対応する別のhtml、および関連グラフィックスがあればその関連グラフィックスを、クライアント・ウェブ・ブラウザのユーザが前記リンクを選択しなくても、前記クライアント・ウェブ・ブラウザの記憶装置に自動的にロードする手段とを含むクライアント・ウェブ・ブラウザ。

【請求項2】前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに応答して、前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込み、前記別のhtmlおよび関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、請求項1に記載のウェブ・ブラウザ。

【請求項3】第1の前記htmlをサーバからロードし、前記第1のhtml、および関連グラフィックスがあればその関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、請求項1に記載のウェブ・ブラウザ。

【請求項4】事前ロードの対象として印を付けられた前記リンクが、第1の前記html中の前記リンクのエントリに事前ロードの対象として印を付けられていることを特徴とする、請求項1に記載のウェブ・ブラウザ。

【請求項5】前記別のhtml、および関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択
30 するのに応答して、前記別のhtml、および関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応するhtmlをロードする手段をさらに含むことを特徴とする、請求項1に記載のウェブ・ブラウザ。

【請求項6】ウェブ・ページ記述子ファイルを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する手段と、事前ロードの対象として印を付けられたリンクの識別に
40 応答して、前記リンクに対応する別のウェブ・ページ記述子ファイル、および関連グラフィックスがあればその関連グラフィックスを、クライアント・ウェブ・ブラウザのユーザが前記リンクを選択しなくても、前記クライアント・ウェブ・ブラウザの記憶装置に自動的にロードする手段とを含むクライアント・ウェブ・ブラウザ。

【請求項7】前記別のウェブ・ページ記述子ファイルおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに応答して、前記別のウェブ・ページ記述子ファイルおよび関連グラフィックスを前記記憶装置から取り込み、前記別

のウェブ・ページ記述子ファイルおよび関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、請求項6に記載のウェブ・ブラウザ。

【請求項8】前記別のウェブ・ページ記述子ファイル、および関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記ウェブ・ページ記述子ファイルに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するの
10 に応答して、前記別のウェブ・ページ記述子ファイルおよび、関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応する第3のウェブ・ページ記述子ファイルをロードする手段をさらに含むことを特徴とする、請求項6に記載のウェブ・ブラウザ。

【請求項9】htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する段階と、事前ロードの対象として印を付けられたリンクの識別に
20 応答して、前記リンクに対応する別のhtml、および関連グラフィックスがあればその関連グラフィックスを、クライアント・コンピュータのユーザが前記リンクを選択しなくても、ワールド・ワイド・ウェブ(WWW)上の遠隔記憶装置から、前記クライアント・コンピュータの記憶装置に自動的にロードする段階とを含むWWWに結合したクライアント・コンピュータを操作する方法。

【請求項10】前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に前記ユーザが前記リンクを選択するのに応答して、前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込む段階と、前記別のhtmlおよび関連グラフィックス
30 に対応するウェブ・ページを表示する段階とをさらに含むことを特徴とする、請求項9に記載の方法。

【請求項11】前記別のhtml、および関連グラフィックスがあれば、その関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが
40 選択するのに応答して、前記別のhtml、および関連グラフィックスがあれば、その関連グラフィックスの事前ロードを停止する段階と、前記別のリンクに対応するhtmlをロードする段階とをさらに含むことを特徴とする、請求項9に記載の方法。

【請求項12】コンピュータ可読媒体と、htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別するようにプロセッサに命令する第1のプログラム命令手段と、前記リンクに対応する別のhtml、および関連グラフィックスがあればその関連グラフィックスを、クライアント・コンピュータのユーザが前記リンクを選択しなく
50 ても、ワールド・ワイド・ウェブ(WWW)上の遠隔記

憶装置から、クライアント・コンピュータの記憶装置に自動的にロードすることによって、事前ロードの対象として印を付けられたリンクの識別に应答するようにプロセッサに命令する第2のプログラム命令手段とを含み、前記第1および第2のプログラム命令手段が前記媒体上に記録されるWWWに結合したクライアント・コンピュータを操作するコンピュータ・プログラム製品。

【請求項13】前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込み、前記別のhtmlおよび関連グラフィックスに対応するウェブ・ページを表示することによって、前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに应答するようにプロセッサに命令する第3のプログラム命令手段をさらに含み、

前記第3のプログラム命令手段が前記媒体上に記録されることを特徴とする、請求項12に記載のプログラム製品。

【請求項14】前記別のhtml、および関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応するhtmlをロードすることによって、前記第2のhtml、および関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するのに应答するようにプロセッサに命令する第4のプログラム命令手段をさらに含むことを特徴とする、請求項12に記載のプログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的にはワールド・ワイド・ウェブ(WWW)上のクライアント・コンピュータに関し、より具体的にはウェブ・ブラウザ・プログラムに関する。

【0002】

【従来の技術】WWWは周知のものであり、多数のコンピュータ・サーバ、それぞれのデータ・ベース、およびクライアント・コンピュータがサーバと通信しデータを要求したりロードしたりすることができるネットワークを含んでいる。サーバは、それ自体のデータ・ベースを直接に管理すること、およびクライアントのユーザに代わって他の遠隔データ・ベースにアクセスすることができる。クライアントは一般に、WWWへのユーザ・インタフェースを提供する「ウェブ・ブラウザ」プログラムを備えている。

【0003】サーバは、「ウェブ・ページ」としてデータをユーザに提示し、各ウェブ・ページは「URL」アドレスによって代表される。URLは、プレフィックスとしてhttpのようなアクセス方式/プロトコル、「ホームページ」とも呼ばれるサーバ名、およびもしあ

ればサフィックスとしてデータ型を含んでいる。サーバ名は一般に、そのサーバを所有する会社、教育機関、またはその他の団体の名称である「ドメイン名」を含んでいる。クライアントがウェブ・ページにアクセスする方法にはいくつかの異なる方法がある。クライアントがサーバ名、およびもしあれば、データ型のサフィックスを知っている場合は、クライアントはサーバに直接、ウェブ・ページを要求することができる。しかし、クライアントがサーバ名しか知らない場合には、クライアントはサーバ名をアドレス指定することができ、これに应答してサーバはこのサーバの「ホームページ」を提示する。少なくともサーバ名をユーザが知らない、これらのウェブ・ページに対しては、キー・ワード・サーチ・エンジンやカタログ・サーチ・エンジンなど、興味を引くサーバまたはウェブ・ページあるいはその両方を識別する各種サーチ・エンジンがある。ホームページおよび他のウェブ・ページは一般に、他のウェブ・ページへのリンクの役目をするテキストまたはグラフィックスを含んでいる。リンクは、「ホット・リンク」または「クリック・ポイント」と呼ばれることもあり、ユーザがマウスでリンクを選択すると、ウェブ・ブラウザはサーバにウェブ・ページを要求し、受領するとそのウェブ・ページを表示する。

【0004】ウェブ・ブラウザがウェブ・ページをサーバに要求するときにはいつでも、ユーザがURLを指定するかまたはリンクを選択するかのもいずれかによって、そのウェブ・ページがhtmlファイルの形でクライアント・マシンにロードされる。htmlファイルは、ウェブ・ページの各コンポーネント、すなわちテキスト、グラフィックス、コンポーネントの性質、各テキストまたはグラフィックが別のウェブ・ページへのリンクであるかどうかなどの仕様を含んでいる。従来のhtmlのフォーマットは業界標準となっており、さらにメアリ E. S. モリス (Mary E. S. Morris) 著「HTML for Fun and Profit」, SunSoft Press, Prentice Hall, 1995等に定義されている。テキストの場合、htmlは実際のテキストおよびそのウェブ・ページ上の位置を定義する。しかし、グラフィックスの場合は、htmlはそのグラフィックス自体を定義するのではなく、代わりに、クライアントから離れた場所にあつてグラフィックスを実際に定義する別のファイルへのポイントを指定する。グラフィックスには複雑なものもある。グラフィックスは、ウェブ・ページに意味を加えリンクとして機能することができる反面、特にグラフィックスが複雑である場合は、ロードするのに時間がかかることがしばしばある。時間遅れは、クライアントとサーバの間の通信回線の遅さに起因する。したがって、リンクを選択した後ユーザは、対応するhtmlおよびグラフィックスがロードされるのを待たなければならず、この待ち時間が非常に長くなることがある。

【0005】

【発明が解決しようとする課題】本発明の一般的な目的は、現在ユーザが見ているウェブ・ページ上のリンクによって参照された別のウェブ・ページへのユーザのアクセスを迅速におこなう方法およびシステムを提供することにある。

【0006】

【課題を解決するための手段】本発明は、ウェブ・ページをロードして表示し、そのウェブ・ページのhtmlを読んで、事前ロードの対象として印を付けられたリンクのエントリを識別するウェブ・ブラウザにある。次いでウェブ・ブラウザは、このリンクに対応する別のhtmlおよび関連グラフィックス・ファイルを記憶装置に自動的に事前ロードする。この自動事前ロードは、ウェブ・ブラウザのユーザがリンクを選択しなくても実施される。ユーザが続いてそのリンクを選択した場合には、対応するhtmlおよび関連グラフィックス・ファイルが局部記憶装置から使用可能となり、そのため、htmlおよび関連グラフィックス・ファイルの取り込みにおける遅延がわずかなる。

【0007】本発明の一特徴に従って、現在表示されているウェブ・ページ中の別のリンクをユーザが選択し、この別のリンクに対応するウェブ・ページが局部記憶装置になく、事前ロードが実行中である場合には、ウェブ・ブラウザはこの事前ロードを停止し、代わりに現在ユーザが選択しているウェブ・ページをロードする。

【0008】

【発明の実施の形態】次に図面を詳細に説明する。全体を通じ同じ参照番号は同じ要素を指している。図1は、通常10で表す、本発明によるクライアント・コンピュータを示したものである。クライアント・コンピュータ10は、ウェブ・ブラウザ・プログラム12およびウェブ・ページを記憶する記憶装置14（RAMおよび/またはディスク）を備えている。ウェブ・ブラウザ12は、ユーザ・インタフェースを提供し、サーバと通信してユーザが要求したウェブ・ページを手に入れ、ウェブ・ページのユーザへの表示を制御する。クライアント10は、モデム、および電話または光ファイバなどの通信回線18を介してWWW16に結合される。WWWは、サーバ20などの多数のサーバを含む。サーバ20は、クライアント10に代わってディスク24のウェブ・ページにアクセスする、またはクライアント10に代わって別のサーバ26にウェブ・ページを要求するウェブ・ページ・アクセス機構プログラム22を有する。図示の例では、クライアント10はhttpアクセス方式を使用してサーバ20と通信する。

【0009】サーバ20がクライアントにウェブ・ページを供給するときには、ウェブ・ページは、http通信に組み込まれたhtmlファイルによって定義される。後により詳細に説明するが、本発明によるhtml

は、このhtmlにリンクしたウェブ・ページのあるものを事前ロードの対象に指定する。ウェブ・ブラウザがhtmlをロードし対応するウェブ・ページを表示すると直ちに、ウェブ・ブラウザは、指定されたリンクに対応するhtmlおよびグラフィックスを記憶装置14に事前ロードし始める。ユーザが続いて、事前ロードされたこれらのウェブ・ページの1つを、現在表示されているウェブ・ページ上のそれぞれのリンクを介して選択した場合には、ウェブ・ブラウザは即座に、局部記憶装置14中のウェブ・ページを取り込み、表示することができる。

【0010】以下は、あるユーザによって設計された本発明に基づくhtmlの例である。

33. <html>

35. <head>

37. <title>Patent Test Page</title>

39. </head>

41. <body>

43. <h3>Test Links</h3>

20 45.

47. Catalog

49. Vacuum

51. Carpet Shampooer

53.

55. </body>

最初のエントリ33の「<html>」は、このファイルがhtmlであることを示す。次のエントリ35の「<head>」は、ページ・ヘッダを示す。次のエントリ37は、以降のテキストが、プリント・ジョブ名およびクイックリストに使用されるウェブ・ページのタイトルであることを示すタグ「<title>」で始まっている。次のエントリ39は、ページ・ヘッダの終わりを示すタグ「</head>」である。次のエントリ41は、htmlページの本体を示す。次のエントリ43は、見出し（heading）を示す。次のエントリ45は、番号なしリストを示す。次のエントリ47は、これがリストの要素であることを意味する「」で始まり、次いで、以降のテキスト「Catalog」をクリックすると指定されたURL「www.test.com」に行くことができることを意味するタグ「<a>」を含んでいる。これは例であり、これによって制限されるものではないが、「Catalog（カタログ）」ウェブ・ページは、現在見ているウェブ・ページを所有するメーカーの、テキストで書かれたカタログである。このウェブ・ページはテキストのみを含みグラフィックスを含まないのでロード時間は短く、このウェブ・ページ的设计者は、このリンクを事前ロードの対象として指定する必要を認めなかった。したがって、エントリ47は標準的な従来技術のhtmlのエントリである。次のエントリ4

9も、リストの要素を意味するで始まり、次いで、以降のテキスト「Vacuum」をクリックすると指定されたURL「www.test1.com」に行くことができることを意味するタグ「<a>」を含んでいる。これは例であり、これによって制限されるものではないが、「Vacuum（掃除機）」ウェブ・ページは、前述のメーカーが販売する掃除機の写真1枚、およびこれについてのその他の情報を含んでいる。したがってロード時間はかなり長くなりそうであり、ウェブ・ページの設計者は、このウェブ・ページを事前ロードの対象とする必要を認めた。本発明に従って、エントリ49は、前述のhtmlに対応するウェブ・ページが表示された後すぐに、URL「www.test1.com」のhtmlを事前ロードしなければならないことを意味するフィールド「PRIORITY = 1」も含んでいる。この事前ロードは、ユーザが要求したり、エントリ49に対応するリンクを選択したりしなくても自動的に実行される。次のエントリ51も、リストの要素を意味するで始まり、次いで、以降のテキスト「Carpet Shampooer」をクリックすると指定されたURL「www.test2.com」に行くことができることを意味するタグ「<a>」を含んでいる。これは例であり、これによって制限されるものではないが、「carpet shampooer（カーペット洗浄機）」ウェブ・ページは、前述のメーカーが販売する洗浄機の写真、およびこれについてのその他の情報を含んでおり、ロード時間はかなり長くなりそうである。したがって、本発明に従ってエントリ51は、前述のhtmlに対応するウェブ・ページが表示された後すぐに、URL「www.test2.com」のhtmlを事前ロードしなければならないことを意味するフィールド「PRIORITY = 10」も含んでいる。この事前ロードは、ユーザが要求したり、エントリ51に対応するリンクを選択したりしなくても自動的に実行される。後により詳細に述べるが、優先度レベルがそれぞれのhtmlを事前ロードする順序を決定する。次のエントリ53は、これが番号なしリストの終わりであることを示す。最後のエントリ55は、これがhtmlの終わりであることを示す。

【0011】図2に、ウェブ・ブラウザ・プログラム12の中のブラウジング・スレッド60による処理を示す。段階62で、ブラウジング・スレッドは、ユーザの選択したウェブ・ページをURL、またはリンクのいずれかによって受け取る。説明の目的上、段階62での最初の選択はURLでおこなわれ、そのURLは前記のhtmlを指すと仮定する。これにตอบสนองして、ブラウジング・スレッドは、このURLをダイナミック・ロード待ち行列63（図1）に加え（段階64）、図3および図4に示すローディング・スレッド66に、ダイナミック・ロード待ち行列を直ちに処理するよう指示するメッセージを送る（段階65）。

【0012】図3および図4は、ローディング・スレッド66の流れ図を構成する。ブラウジング・スレッド6

0からメッセージを受け取ると（段階82）、ローディング・スレッドは最初に、このメッセージは、ダイナミック・ロード待ち行列の処理に関するもので、事前ロード待ち行列73（図1）の処理に関するものではないと判断する（判断84および判断130）。ローディング・スレッドは次に、ユーザが選択したウェブ・ページが以前の事前ロード操作の結果として記憶装置14に現在あるかどうかを判断する（判断104）。（事前ロード操作を以下に説明する。）もしあれば、ローディング・スレッドは、記憶装置14からウェブ・ページを取り込み、表示する（段階106）。記憶装置14は局部記憶装置なので、この操作は迅速に実行される。しかし、ユーザが選択したウェブ・ページが記憶装置14に現在ない場合には、ローディング・スレッドは、事前ロード操作に従って現在事前ロード処理中のウェブ・ページがあるかどうかを判断する（判断110）。もしなければ、ローディング・スレッドは、ユーザが選択したウェブ・ページをサーバに要求し、ユーザが選択したウェブ・ページを記憶装置14にロードし、ユーザが選択したウェブ・ページを表示する（段階112）。判断110に戻る。事前ロード操作に従って事前ロード処理中のウェブ・ページが現在ある場合には、ローディング・スレッドは、このページがユーザが選択したウェブ・ページかどうかを判断する（判断113）。もしそうであれば、ローディング・スレッドは、ユーザが選択したウェブ・ページのロードを完了させ、ユーザが選択したウェブ・ページを表示する（段階114）。しかし、ユーザが選択したウェブ・ページ以外のウェブ・ページが現在事前ロードされている場合は、ローディング・スレッドは、この別のウェブ・ページの事前ロードを停止させ（段階120）、ユーザの選択したウェブ・ページをサーバに要求してロードし、ユーザの選択したウェブ・ページを表示する（段階112）。段階106、112、114のいずれかの後、ローディング・スレッドは、ダイナミック・ロード待ち行列上のウェブ・ページのロードおよび表示に成功したことを、ブラウジング・スレッドに通知する（段階124）。

【0013】ローディング・スレッドから表示の通知を受け取ると（段階67）、ブラウジング・スレッドは、現在表示されているウェブ・ページのhtmlを読み、現在表示されているウェブ・ページ中であって、事前ロードを指示する「priority」フラグを有する全てのリンクを探し出す（段階68）。前記のhtmlには、このようなリンクが2つ、エントリ49および51によって定義されている。しかし、ブラウジング・スレッドは最初に、エントリ47によって定義された、事前ロードの対象としての印を付けられていないリンクに出会う。したがって、エントリ47に出会ったときには、判断70から、一切の事前ロード操作を迂回して判断72に移り、判断72からは段階68に再び戻って次のエントリ

49をチェックする。この繰返しの間に、ブラウジング・スレッド60は、事前ロードを指示する「priority」フラグで印を付けられたエントリ49に出会う（判断70）。その結果、ブラウジング・スレッドは、エントリ49中のURLを、事前ロード待ち行列73の優先度レベルに応じた位置に加える（段階74）。優先度レベルが高いほど、URLは、待ち行列中の先頭に近い位置に置かれる。段階68および70の次の繰返しの間に、ブラウジング・スレッドは、事前ロードを指示する「priority」フラグをやはり含んだエントリ51に出会い、そのブラウジング機能が、エントリ51中のURLを事前ロード待ち行列73に加える（段階74）。エントリ51が「priority 10」の印を付けられ、エントリ49が「priority 1」の印を付けられている図示の例では、エントリ51のURLは、事前ロード待ち行列中でエントリ49のURLの前に置かれる。ブラウジング・スレッドは前記html中の全てのリンクのチェックを完了したので、ブラウジング・スレッドはローディング・スレッド66に事前ロード待ち行列を直ちに処理するよう通知する（段階76）。

【0014】ローディング・スレッド66は、ブラウジング・スレッドから事前ロード待ち行列を処理する通知メッセージを受け取り（段階82および判断84）、これに応答して、事前ロード待ち行列上の最初のURLを判断する（判断86）。次いで、ローディング・スレッドは、対応するhtmlおよび関連グラフィックスをサーバに要求する。ローディング・スレッドがhtmlおよび関連グラフィックスを受け取り、ロードする（段階88）間にも、ローディング・スレッドは、ブラウジング・スレッドからのメッセージを定期的にチェックする。メッセージを受け取っていない場合（判断90）、ローディング・スレッドはロードを継続する。すなわち、ロードすべき現在のhtmlおよび関連グラフィックスのデータがさらにあるかどうかを判断し（判断92）、もしあれば、ループは段階88に戻って、htmlまたは関連グラフィックスの次の部分をロードする。しかし、ローディング・スレッドが現在のhtmlおよび関連グラフィックスの最後の部分をロードしてしまっている場合は、ローディング・スレッドは、事前ロード待ち行列に別のURLがあるかどうかを判断し（段階94および判断86）、ループは判断86に戻って、この別のURLを読む。表示されたウェブ・ページが事前ロードされるべきウェブ・ページを2つ含む前述の例では、事前ロード待ち行列には少なくとも2つのエントリがある。

【0015】判断90に戻る。ユーザがリンクを選択し、直ぐに表示することを指示するメッセージをローディング・スレッドがブラウジング・スレッドから受け取った場合には（判断90）、ローディング・スレッドは、ユーザが、現在記憶装置14にあるウェブ・ページ

を選択したかどうかを判断する（判断104）。もしそうであれば、ローディング・スレッドは記憶装置14からウェブ・ページを取り込み、表示する（判断106）。しかし、ユーザが選択したウェブ・ページが現在記憶装置14にない場合には、ローディング・スレッドは、事前ロード操作に従ってロード処理中のウェブ・ページがあるかどうかを判断する（判断110）。もしなければ、ローディング・スレッドは、ユーザが選択したウェブ・ページをサーバに要求し、ユーザが選択したウェブ・ページを記憶装置14にロードし、ユーザが選択したウェブ・ページを表示する（段階112）。判断110に戻る。事前ロード操作に従って事前ロード処理中のウェブ・ページが現在ある場合には、ローディング・スレッドは、このページがユーザが選択したウェブ・ページかどうかを判断する（判断113）。もしそうであれば、ローディング・スレッドは、ユーザが選択したウェブ・ページのロードを完了させ、ユーザが選択したウェブ・ページを表示する（段階114）。しかし、ユーザが選択したウェブ・ページ以外のウェブ・ページが現在事前ロードされている場合は、ローディング・スレッドは、この別のウェブ・ページの事前ロードを停止させ（段階120）、ユーザの選択したウェブ・ページをサーバに要求してロードし、ユーザの選択したウェブ・ページを表示する（段階112）。

【0016】段階106、112、または114の後、ローディング・スレッドは、ユーザが選択した、直ちに表示すべきウェブ・ページが表示されたことを、ブラウジング・スレッドに通知する。また、ローディング・スレッドは、事前ロードがまだ完了していないページが事前ロード待ち行列にあるかどうかを判断する（判断122）。もしあれば、ローディング・スレッドは判断92の処理を継続し、なければ、ローディング・スレッドは終了となる。

【0017】前述の事項に基づいて、本発明によるウェブ・ブラウザを開示した。しかし、多数の修正および置換えは、本発明の範囲から逸脱することなく実施することができる。希望する場合には、例えば、事前ロードを全く止める、または、優先度レベルが5未満の全ての事前ロードを止めるなど、事前ロードの優先度レベルが低い事前ロードを止める選択をユーザが任意にできる機能を、ウェブ・ブラウザに与えることもできる。したがって、本発明は、これによって制限されない例示によって開示されたものであり、本発明の範囲を決定する特許請求の範囲を参照すべきものである。

【0018】まとめとして、本発明の構成に関して以下の事項を開示する。

【0019】(1) htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する手段と、事前ロードの対象として印を付けられたリンクの識別に応答して、前記リンクに対応する別のhtml、お

および関連グラフィックスがあればその関連グラフィックスを、クライアント・ウェブ・ブラウザのユーザが前記リンクを選択しなくても、前記クライアント・ウェブ・ブラウザの記憶装置に自動的にロードする手段とを含むクライアント・ウェブ・ブラウザ。

(2) 前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに応答して、前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込み、前記別のhtmlおよび関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、上記(1)に記載のウェブ・ブラウザ。

(3) 第1の前記htmlをサーバからロードし、前記第1のhtml、および関連グラフィックスがあればその関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、上記(1)に記載のウェブ・ブラウザ。

(4) 事前ロードの対象として印を付けられた前記リンクが、第1の前記html中の前記リンクのエントリに事前ロードの対象として印を付けられていることを特徴とする、上記(1)に記載のウェブ・ブラウザ。

(5) 前記別のhtml、および関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するのに応答して、前記別のhtml、および関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応するhtmlをロードする手段をさらに含むことを特徴とする、上記(1)に記載のウェブ・ブラウザ。

(6) ウェブ・ページ記述子ファイルを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する手段と、事前ロードの対象として印を付けられたリンクの識別に応答して、前記リンクに対応する別のウェブ・ページ記述子ファイル、および関連グラフィックスがあればその関連グラフィックスを、クライアント・ウェブ・ブラウザのユーザが前記リンクを選択しなくても、前記クライアント・ウェブ・ブラウザの記憶装置に自動的にロードする手段とを含むクライアント・ウェブ・ブラウザ。

(7) 前記別のウェブ・ページ記述子ファイルおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに応答して、前記別のウェブ・ページ記述子ファイルおよび関連グラフィックスを前記記憶装置から取り込み、前記別のウェブ・ページ記述子ファイルおよび関連グラフィックスに対応するウェブ・ページを表示する手段をさらに含むことを特徴とする、上記(6)に記載のウェブ・ブラウザ。

(8) 前記別のウェブ・ページ記述子ファイル、および

関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記ウェブ・ページ記述子ファイルに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するのに応答して、前記別のウェブ・ページ記述子ファイルおよび、関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応する第3のウェブ・ページ記述子ファイルをロードする手段をさらに含むことを特徴とする、上記(6)に記載のウェブ・ブラウザ。

(9) htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別する段階と、事前ロードの対象として印を付けられたリンクの識別に応答して、前記リンクに対応する別のhtml、および関連グラフィックスがあればその関連グラフィックスを、クライアント・コンピュータのユーザが前記リンクを選択しなくても、ワールド・ワイド・ウェブ(WWW)上の遠隔記憶装置から、前記クライアント・コンピュータの記憶装置に自動的にロードする段階とを含むWWWに結合したクライアント・コンピュータを操作する方法。

(10) 前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に前記ユーザが前記リンクを選択するのに応答して、前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込む段階と、前記別のhtmlおよび関連グラフィックスに対応するウェブ・ページを表示する段階とをさらに含むことを特徴とする、上記(9)に記載の方法。

(11) 前記別のhtml、および関連グラフィックスがあれば、その関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するのに応答して、前記別のhtml、および関連グラフィックスがあれば、その関連グラフィックスの事前ロードを停止する段階と、前記別のリンクに対応するhtmlをロードする段階とをさらに含むことを特徴とする、上記(9)に記載の方法。

(12) コンピュータ可読媒体と、htmlを読み、事前ロードの対象として印を付けられたリンクのエントリを識別するようにプロセッサに命令する第1のプログラム命令手段と、前記リンクに対応する別のhtml、および関連グラフィックスがあればその関連グラフィックスを、クライアント・コンピュータのユーザが前記リンクを選択しなくても、ワールド・ワイド・ウェブ(WWW)上の遠隔記憶装置から、クライアント・コンピュータの記憶装置に自動的にロードすることによって、事前ロードの対象として印を付けられたリンクの識別に応答するようにプロセッサに命令する第2のプログラム命令手段とを含み、前記第1および第2のプログラム命令手段が前記媒体上に記録されるWWWに結合したクライアント・コンピュータを操作するコンピュータ・プログラ

ム製品。

(13) 前記別のhtmlおよび関連グラフィックスを前記記憶装置から取り込み、前記別のhtmlおよび関連グラフィックスに対応するウェブ・ページを表示することによって、前記別のhtmlおよび関連グラフィックスが前記記憶装置に事前ロードされた後に、前記ユーザが前記リンクを選択するのに応答するようにプロセッサに命令する第3のプログラム命令手段をさらに含み、前記第3のプログラム命令手段が前記媒体上に記録されることを特徴とする、上記(12)に記載のプログラム製品。

(14) 前記別のhtml、および関連グラフィックスがあればその関連グラフィックスの事前ロードを停止し、前記別のリンクに対応するhtmlをロードすることによって、前記第2のhtml、および関連グラフィックスがあればその関連グラフィックスが事前ロードされている間に、第1の前記htmlに対応する現在表示されているウェブ・ページ中の別のリンクをユーザが選択するのに応答するようにプロセッサに命令する第4のプログラム命令手段をさらに含むことを特徴とする、上記(12)に記載のプログラム製品。

【図面の簡単な説明】

【図1】ウェブ・ブラウザを備えた本発明によるクライ

アント・コンピュータ、サーバ、およびサーバにクライアント・コンピュータを相互接続するネットワークのブロック図である。

【図2】図1のウェブ・ブラウザ内のブラウジング・スレッドによる処理を示した流れ図である。

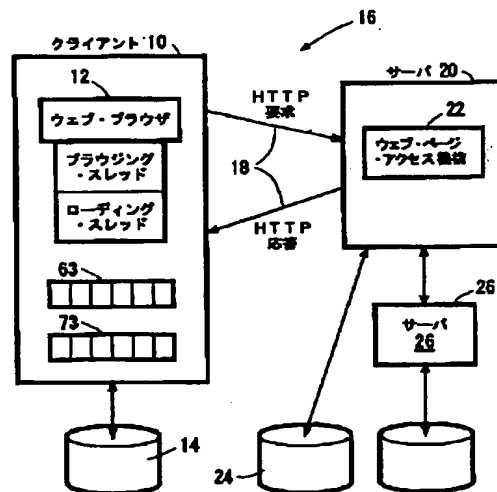
【図3】図1のウェブ・ブラウザ内のローディング・スレッドによる処理を示した流れ図である。

【図4】図1のウェブ・ブラウザ内のローディング・スレッドによる処理を示した流れ図である。

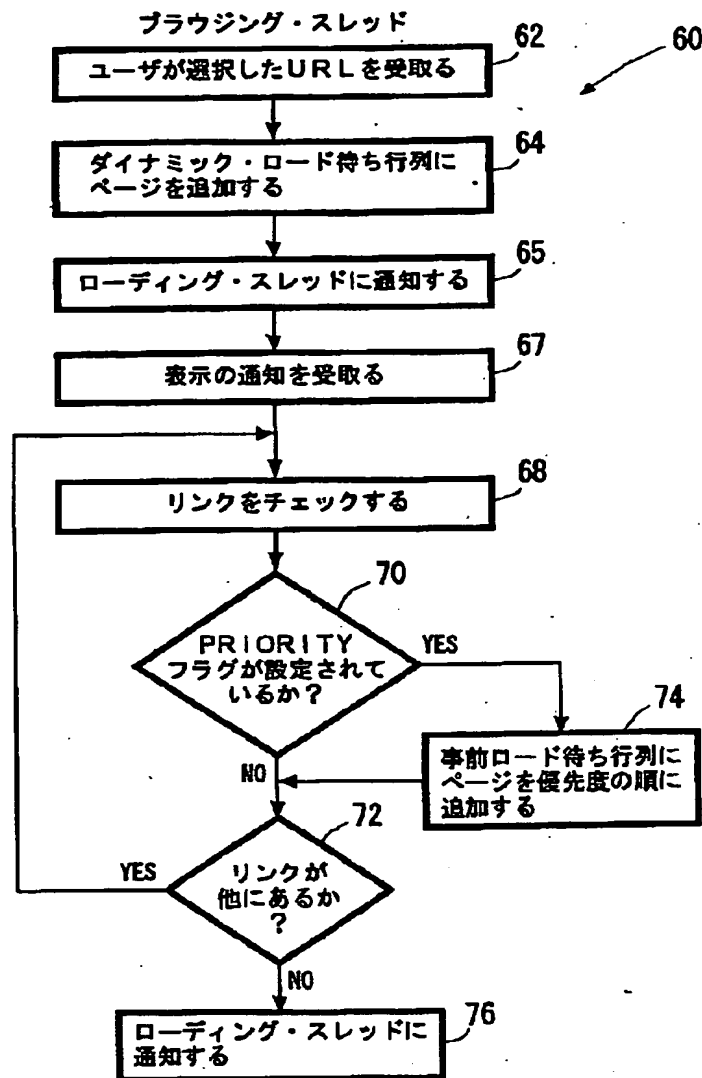
【符号の説明】

- 10 クライアント・コンピュータ
- 12 ウェブ・ブラウザ・プログラム
- 14 記憶装置
- 16 ワールド・ワイド・ウェブ (WWW)
- 18 通信回線
- 20 サーバ
- 22 ウェブ・ページ・アクセス機構プログラム
- 24 ディスク
- 26 サーバ
- 60 ブラウジング・スレッド
- 63 ダイナミック・ロード待ち行列
- 66 ローディング・スレッド
- 73 事前ロード待ち行列

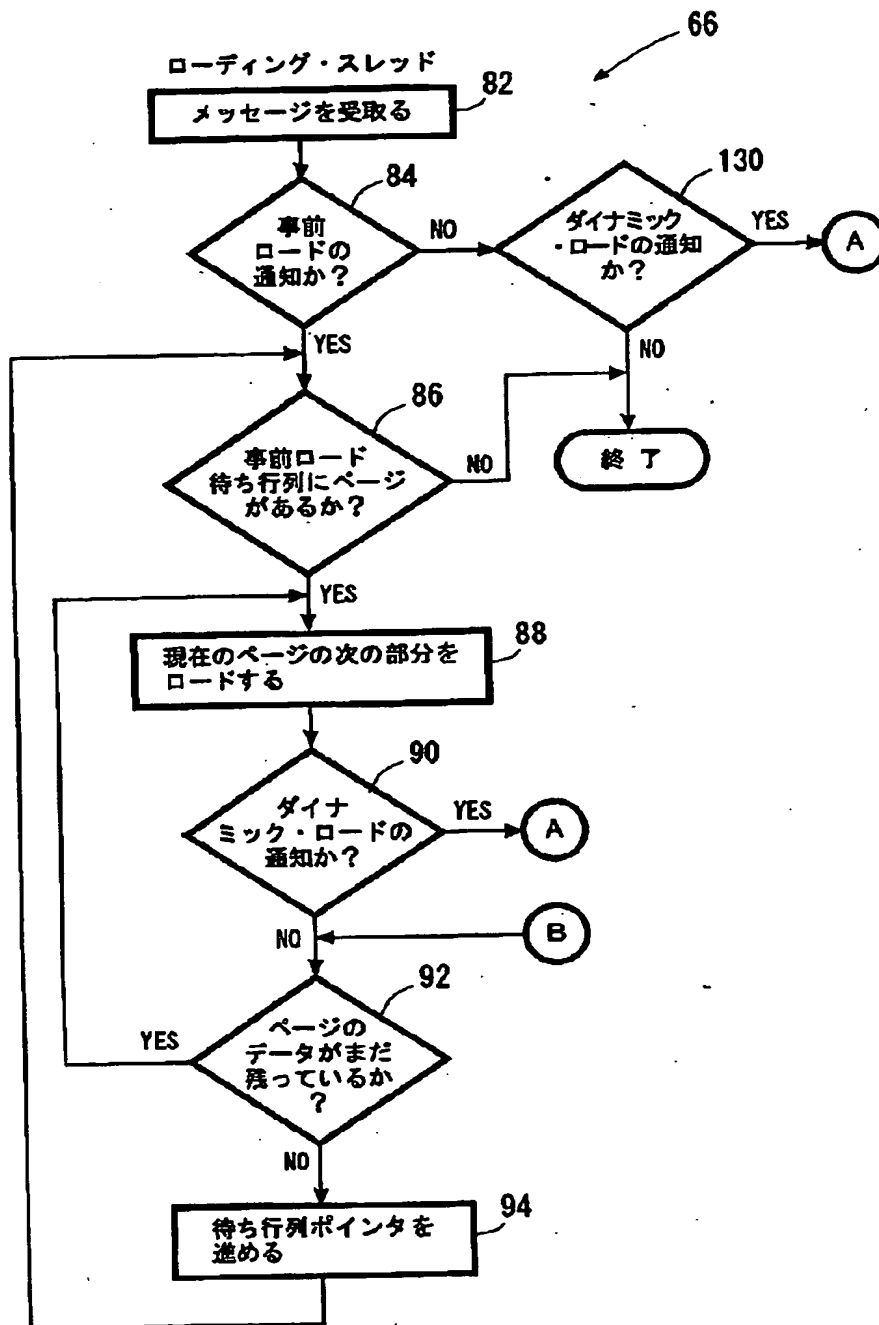
【図1】



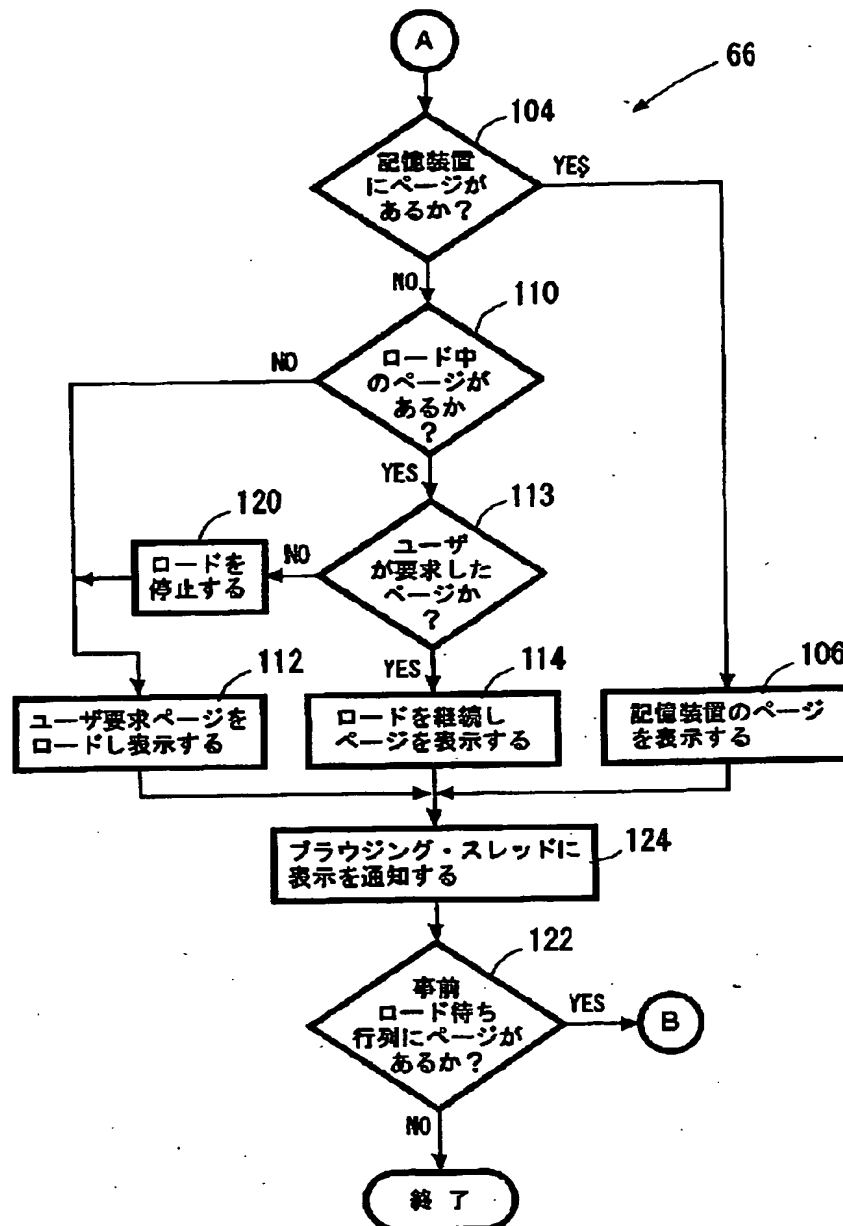
【図2】



【図3】



【図4】



フロントページの続き

(72)発明者 マイケル・スローン・ボマー
 アメリカ合衆国13760 ニューヨーク州エ
 ンディコット モス・アヴェニュー 307

(72)発明者 ウィリアム・フランシス・フィリップス
 アメリカ合衆国13827 ニューヨーク州オ
 ウェゴウッドヘブン・ドライブ 599